

UNITED STATES DISTRICT COURT
DISTRICT OF MASSACHUSETTS

SCANSOFT, INC.,

Plaintiff,

v.

VOICE SIGNAL TECHNOLOGIES, INC.,
LAURENCE S. GILLICK, ROBERT S.
ROTH, JONATHAN P. YAMRON, and
MANFRED G. GRABHERR,

Defendants.

C.A. No. 04-10353-PBS

AFFIDAVIT OF CHARLES C. WOOTERS

I, Charles C. Wooters, on oath, depose and say as follows:

1. I am a research scientist in the field of speech recognition, and am a Senior Research Engineer at the International Computer Science Institute, Berkeley, California. I hold a Ph.D. in Speech Recognition from the University of California, Berkeley. I also hold a Master's degree in Linguistics from the University of California, Berkeley. I have published numerous articles concerning various methods and techniques in the field of speech recognition, and am a named inventor on a number of patent applications. My curriculum vitae is attached as Exhibit A to this Affidavit.

2. I make this Affidavit in support of Voice Signal Technologies, Inc.'s Memorandum in Support of its Objections to the Magistrate Judge's Order Regarding Trade Secrets, and to assist the Court in understanding the general topic areas described as trade secrets by ScanSoft.

3. In general, speech recognition involves receiving sounds from a human speaker, matching those sounds against a database of known combinations of sounds that correspond to words (or a mathematical model of those sounds) and, using statistical algorithms, choosing the most probable words spoken. The next few paragraphs provide a brief overview of one way in which such a system may be constructed.

4. There are many levels of matching that occur simultaneously in a speech recognition system. At a low level, the matching occurs with sub-word units. These sub-word units or “phonemes” are simply the basic sounds of the language. For example, the word “cat” consists of three phonemes: “k”, “ae”, and “t”. Each phoneme is matched against a database of phoneme models that were constructed from recordings taken from various human subjects.

5. Statistical data may also be applied at the word level, to inform the matching process. For example, by mining the English language electronic media (e.g. *The Wall Street Journal*, *The New York Times*, etc.), statistical data is collected on the relative positioning of words in English language usage. This may, for example, inform the system that if it recognizes an adjective (e.g. “purple” or “beautiful”), the next word is likely to be a noun (e.g. “balloon”).

6. I have reviewed a copy of ScanSoft, Inc.’s Response to Voice Signal’s Second Set of Interrogatories. ScanSoft’s response to Interrogatory No. 1 lists six sub-categories of alleged “trade secrets.” The Responses list general topic areas well-known in the speech recognition field and do not specify or particularize discrete techniques which, upon review of the Responses, can be said to be unique or proprietary. Each subcategory is addressed below.

a. Subcategory (1) of ScanSoft’s response to Interrogatory No. 1 is “proprietary techniques for duration modeling of speech.” No particular technique for duration modeling is described.

Duration modeling is a common technique in speech recognition. The idea behind duration modeling is that the recognizer can use information about the duration, or length, of what was spoken to help it figure out what was said. For example, the word “of” has a much shorter duration than the word “offensive”. If the recognizer receives an utterance with a short duration, it is unlikely to be the word “offensive” and thus it can eliminate this long word from its list of possible recognition candidates.

One technique for duration modeling is to model the duration of a given word via the sounds, or phonemes, that comprise the word. For example, the duration of the word “cat” may be modeled by summing the durations for the sounds “k”, “ae”, and “t”. Alternatively, the model may focus on the duration of the entire word and may be modeled by collecting several spoken samples of the word “cat” and computing the average duration of the word “cat” over all of the samples. Speech recognition systems may create duration models not just for phonemes and words, but also for sentences, phrases, parts of phonemes, etc. There are many different known techniques that one can use and combine to implement duration modeling in a speech recognition system. There are also many different speech units (phonemes, words, phrases, sentences, etc.) that one could choose to model. Subcategory (1) of ScanSoft’s response does not describe any particular duration model nor does it specify the speech unit over which the duration modeling is applied.

b. Subcategory (2) of ScanSoft’s response to Interrogatory No. 1 is “specific methods for organizing, categorizing and interpreting word sequence hypotheses.” No “specific method” is identified in Subcategory (2).

“Word sequence hypotheses” is a general description of a common speech recognition component. A “word sequence hypothesis” refers to any sequence of words that may have been

generated by a statistical model of the language, and such models are often based on publicly available data. It can be at the level described above, i.e. that adjectives are more likely to be followed by nouns than by verbs. It can also be specific to commonly used word sequences, such as “President Bush spoke” or “stocks were up”. Word sequence hypotheses are common to all continuous speech recognition systems. All continuous speech recognizers require methods for organizing, categorizing and interpreting a word sequence hypothesis in order to function. Methods for organizing, categorizing and interpreting word sequence hypotheses are expressed as algorithms and implemented in source code. ScanSoft does not identify any specific method which is claimed by ScanSoft to be novel, unique or proprietary.

c. Subcategory (3) of ScanSoft’s response to Interrogatory No. 1 is “speech recognition architecture, including specific structural details such as acoustic matching, phoneme look ahead, lexical tree pre-filtering, word matching and scoring via a router.”

The term “speech recognition architecture” is a general term that refers to the organization or layout of the various component parts of a speech recognition system. These component parts can be thought of as building blocks that can be arranged in different ways to construct a speech recognizer. The terms listed in Subcategory (3) each are general descriptions of common building blocks.

The term “acoustic matching” refers to the part of a speech recognition system that attempts to match pre-recorded speech sounds with sounds or words spoken by a user. Every speech recognizer must have a component that performs acoustic matching. Without such a component, the recognizer would be “deaf,” (i.e., it could not perform the first step of speech recognition). There are many techniques that may be used to implement the acoustic match component. For example, two common approaches are: artificial neural networks and hidden

markov models. Both are the subject of numerous publications in the field of speech recognition. Thus, the term “acoustic matching” refers to a group of techniques and not to one specific proprietary technique. No unique or allegedly proprietary aspect of a particular acoustic matching technique, is disclosed in response to Interrogatory No. 1, Subcategory (3).

The term “phoneme look ahead” refers to a technique, common in the field of speech recognition, that is used for guiding the search of the speech recognizer. As explained above, speech recognition systems operate by receiving from a user a series of sounds, searching through a vast number of possible utterances or words and trying to determine the most probable word or words that were spoken. Phoneme look ahead helps the speech recognizer by providing a general idea about the phonemes (or sounds) that are present in the incoming speech. For example, the phoneme look ahead may say the next sound is a vowel. Because it is working quickly, it will not necessarily identify which exact vowel was spoken, but just knowing that the next sound is a vowel (as opposed to a consonant) helps the recognizer to eliminate certain recognition candidates. This guides the recognizer in its search and reduces the amount of computation needed. No unique or allegedly proprietary implementation of this technique was specified by ScanSoft.

The term “lexical tree pre-filtering” refers to a technique to help reduce the amount of computation performed in a speech recognition system. The key idea here is that a large list of words can be organized into a tree-like structure to take advantage of the fact that many words share a common “prefix.” For example, “boat”, “boy”, and “bark” all share the “prefix” “b”, and might schematically, be organized in a tree. Once the recognizer has concluded that what was received has the prefix “b” the probability that a word has been spoken that does not begin with that prefix is significantly reduced. By organizing the list of words into a lexical tree, the

amount of computation that has to be done by the recognizer can be greatly reduced. Lexical trees are commonly used in speech recognizers. Lexical tree pre-filtering uses lexical trees to pre-select words that look like promising candidates for evaluation by the speech recognizer and to eliminate words that are unlikely to match. No unique or allegedly proprietary implementation of this technique was specified by ScanSoft.

The phrase “word matching and scoring via a router” does not refer to any technique of which I am aware. The sub-phrase “word matching and scoring” describes the function performed by all speech recognition systems -- they match spoken sounds against stored acoustic patterns that correspond to phonemes, words or phrases and produce a score for each match. Typically, the word, phrase or phoneme with the best score is selected. To perform this function via a router implies some sort of distributed speech recognition. These techniques are well known in the field and are not unique or proprietary to ScanSoft.

d. Subcategory (4) of ScanSoft’s response to Interrogatory No. 1 is “specific proprietary language model implementations, including the language models selected and their organization and interaction.”

A language model is used to compute the likelihood of a sequence of words in an utterance. Language models are created by analyzing large collections of text. The language model estimates the probability that a particular word was spoken, based upon the other words around it (typically the words immediately preceding it). For instance, in actual usage, the words “United States District” are more likely to be followed by the word “court” than “cart”. A language model may contain the probabilities associated with both and would be one of the components in the recognizer’s scoring/matching process. The computation of these language model probabilities is a common operation in speech recognition systems.

Often, depending on the application, a speech recognition system may store several language models and switch between them dynamically based on interactions with a user. The selection and organization of language models within a speech recognition system can be done in many different ways. For example, a language model may be represented as a “finite state grammar,” or it may be represented as a “weighted graph.” ScanSoft’s Interrogatory No. 1, Subcategory (4) claims a “specific proprietary... implementation” of a language model, but it neither describes a specific modeling component or technique, nor discloses what is said by ScanSoft to be unique or proprietary.

e. Subcategory (5) of ScanSoft’s response to Interrogatory No. 1 is “the use of mixture models consisting of phoneme elements and genones and the probabilities assigned to each sound.” No specific “mixture model” is identified in Subcategory (5).

A central issue that must be dealt with when building a speech recognition system is the data sparsity problem. Speech recognition systems are “data hungry” -- the more data that is used to train the system, the better the accuracy. Additionally, speech recognition systems typically perform better when they model very specific sounds (phonemes) of the language. For example, notice the difference in the position of the lips when forming the “k” sound in “key” versus the “k” sound in “coo”. For the word “key” the lips are pulled back, but for “coo” the lips are rounded. The acoustic realizations of these two versions of “k” are slightly different. This difference arises due to the influence of the other sounds in the word: the “ey” and the “oo”. Human ears are generally not sensitive to these small differences, but computers are. So, in this example, it would be better for a speech recognition system if it could use two different models of the “k” sound.

Unfortunately, using context-specific models of sounds results in an increase in the number of models that have to be trained. In our “key” / “coo” example, we have two “k” models to train instead of just one, and each model will require its own set of training data. So, as we increase the number of specific models of speech sounds, we need more data to train these models. This issue sets up a trade-off between the specificity of the models used by the speech recognizer, and the amount of training data needed to train those models.

Mixture modeling is a common technique in speech recognition that is used to make adjustments in the trade-off mentioned above. The idea is to model a speech sound (a “phoneme”) as a mixture of small pieces of sound. This mixture can be used to assign a probability, or score, to the phoneme model. Typically, we create a large pool of these small pieces and all of the different phoneme models share the sounds in the pool. It works much like commuter car pooling which reduces the number of cars on the road by having several people share a single car. This helps to reduce the overall amount of data needed for training the speech recognition system.

A genome is a particular method of setting up the sharing arrangement between the phoneme models and the small pieces of sounds in the pool. In our example above, the genome selects the data (the “k” sounds) from which the model for the word “key” will be built. The genome method is well-known in the speech recognition community and was developed by researchers at SRI’s (formally known as Stanford Research Institute International) speech recognition research lab.

The use of mixture models consisting of phoneme elements and genomes is a well-known topic of published research in speech recognition. Interrogatory Response No. 1, Subcategory (5) does not describe the particular mixture model which is said to be unique (nor is

anything said about what might be unique about ScanSoft's use of phoneme elements and genones).

f. Subcategory (6) of ScanSoft's response to Interrogatory No. 1 is "proprietary methods of using look up tables for score computation and other purposes." No "proprietary method" is described.

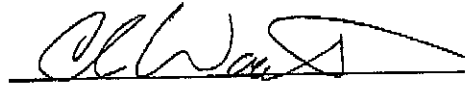
The use of look up tables is a well-known speedup technique used in implementing speech recognizers (and for other applications in computer science generally) whereby a table of previously calculated results is stored in memory to be accessed at a later time, rather than computing these results in real time. There are many publicly-available look up tables which can be used in speech recognition. The mere use of a look up table in speech recognition is a standard method and is not itself proprietary. ScanSoft's response does not identify the lookup table used and does not identify what it claims to be proprietary about that look up table.

Signed under the pains and penalties of perjury this _____ day of _____, 2005.

Charles C. Wooters

3912054v1

Signed under the pains and penalties of perjury this 13th day of April, 2005.

A handwritten signature in black ink, appearing to read "C. Wooters", written over a horizontal line.

Charles C. Wooters